



Minimal Surfaces for Stereo

Citation

Buehler, Chris, Steven J. Gortler, Michael F. Cohen, and Leonard McMillan. 2002. Minimal surfaces for stereo. In Proceedings of the Seventh European Conference on Computer Vision (ECCV), Copenhagen, Denmark, May 28-31, 2002, ed. European Conference on Computer Vision and Anders Heyden, 1-14. Lecture Notes In Computer Science 2352. Berlin: Springer.

Published Version

http://dx.doi.org/10.1007/3-540-47977-5_58

Permanent link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:2634284>

Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at <http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA>

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. [Submit a story](#).

[Accessibility](#)

Minimal Surfaces for Stereo

Chris Buehler	Steven J. Gortler	Michael F. Cohen	Leonard McMillan
MIT/LCS	Harvard University	Microsoft Research	MIT/LCS

Abstract. Determining shape from stereo has often been posed as a global minimization problem. Once formulated, the minimization problems are then solved with a variety of algorithmic approaches. These approaches include techniques such as dynamic programming min-cut and alpha-expansion. In this paper we show how an algorithmic technique that constructs a discrete spatial minimal cost surface can be brought to bear on stereo global minimization problems. This problem can then be reduced to a single min-cut problem. We use this approach to solve a new global minimization problem that naturally arises when solving for three-camera (trinocular) stereo. Our formulation treats the three cameras symmetrically, while imposing a natural occlusion cost and uniqueness constraint.

1 Introduction

Determining shape from stereo has often been posed as a global minimization problem [2, 13]. In these formulations one solves for the shape that best predicts the observed image data and is consistent with a set of prior assumptions. These priors often penalize discontinuities and occlusions in the solutions. Once formulated, the minimization problems are then solved with a variety of algorithmic approaches. These approaches include techniques such as dynamic programming [8], min-cut [15, 11], and alpha-expansion [19].

In this paper we show how an algorithmic technique that constructs a discrete spatial minimal cost surface (which we will abbreviate MS for minimal surface) can be brought to bear on stereo global minimization problems. We first show that finding a minimal cost surface within a discrete spatial complex is a natural generalization to the problem of finding a minimal cost path in a planar graph. We then describe how the global MS can be found in polynomial time by solving for the min-cut over the dual graph of the input spatial complex. As an example of its utility, we show how the MS approach can be used to easily derive the min-cut graph problem used by Ishikawa and Geiger to solve a two-camera stereo problem [11].

We extend the MS approach to solve a new global minimization problem that naturally arises when solving for trinocular stereo. Our formulation treats the three cameras symmetrically, while imposing a natural occlusion cost and uniqueness constraint. This has not been achieved in previous algorithms utilizing more than two cameras. To aid our formulation, we also utilize trinocular rectification [1] in a novel way.

2 Previous work

There have been many approaches in the literature to formulate stereo as a global minimization problem. Each approach makes slightly different choices as to the space of

	gly95	chmr96	ig98	rc98	bvz98	kz01	MS
scanline/fullImage	sl	sl ¹	fi	fi	fi	fi	fi
number of cams	2	2	2	any	any	2	3
uniqueness	y	y	y	n	n	y	y
occlusion penalty	y	y	y	n	n	y	y
monotonicity	y	y	y	n	n	n	y
smoothness penalty	general	count+zigzag	count	count ²	general	zigzag ³	count
polynomial time	y	y	y	y	n	n	y

Table 1. Comparison of algorithms according our desired criteria. Notes: 1. chmr96 also describes a full image generalization using annealing methods. 2. rc98 scales each of the counted disparity jumps by the local matching cost. 3. kz01 also investigate a fully general smoothness penalty but find that it leads to NP-Complete subproblems.

allowed solutions and the functional minimized. These different formulations also give rise to different algorithmic solutions, some of which run in polynomial time and some of which are NP-Complete. To help put our algorithm in proper context we compare some of the highlights from the literature along a variety of axes (see Table 1).

Scanline vs. full image Some algorithms operate on each scanline independently, while others treat the entire image as a single problem. Single scanline approaches include the work by [8, 9]. The advantage of the scanline approach is that the associated minimization problems can be solved exactly and efficiently using dynamic programming techniques. Generalizing dynamic programming beyond the single scanline has proven quite difficult, and many early “full image” formulations appealed to approximation methods (such as simulated annealing) [8]. Other full image formulations result in NP-Complete problems [20, 12]. The only full-image formulations that have exact and efficient solutions have used min-cut techniques [15, 11].

Scanline formulations originally solved using dynamic programming approaches could also have been solved using shortest-path approaches. While the concept of dynamic programming does not generalize well to higher dimensions, we show that the idea of shortest path does generalize to higher dimensions as a minimal cost surface problem, which can be solved efficiently using min-cut techniques. This is the basis for our full image approach.

Uniqueness If each pixel represents a single point in space, then a uniqueness constraint should enforce that a pixel in one camera does not match more than one pixel in a different camera. Exceptions are sometimes made as in the so called tilted-plane solution [11]. When more than two cameras are used, there may be data from a subset of cameras that support a solution with multiple surface elements on the line of sight of one pixel [16]. In our work we enforce the uniqueness constraint.

Occlusion Many formulations penalize pixels in one camera that are unmatched with any pixel in other cameras, justified with Bayesian arguments as in [9, 3]. We use a similar occlusion penalty.

Monotonicity Many of the two-camera approaches have included monotonicity constraints (i.e., features do not change their left-right ordering between the left and right images). Although most appropriate for small baseline configurations, monotonicity together with the uniqueness constraint facilitates the use of dynamic programming approaches. One notable exception is the work of [12] where the monotonicity constraint is lifted. Unfortunately, their very general formulation leads to an NP-Complete problem. In our work we show how monotonicity can be generalized naturally to the three-camera setting as part of an efficient MS solution.

Number of cameras Many previous formulations restrict themselves to two cameras. Methods that use three or more cameras in general position can be expected to be more robust since they can exploit both horizontal and vertical parallax. Unfortunately, the multi-camera approaches [15, 20] have difficulty in penalizing occlusions and enforcing *uniqueness* in the input cameras. In other words, their solutions do not penalize unmatched pixels, and may allow for a pixel in one camera to match numerous pixels in other cameras.

In our work we show how a two-camera formulation can be generalized naturally to three cameras in general position, while still penalizing occlusions and enforcing uniqueness.

Smoothness Most previous formulations define a penalty for non-smoothness in the resulting shape, again justified with Bayesian arguments [9, 3]. The various formulations differ greatly in how non-smoothness is penalized.

In most cases non-smoothness is measured by summing the absolute value of the disparity jumps between neighboring matches [11]. Given uniqueness and monotonicity constraints, this penalty is the equivalent to an occlusion penalty; the only way to create a neighboring match with a disparity difference of k is have k unmatched pixels (see Figure 1). Cox [8] also adds a “zig zag” penalty to favor solutions with a few large discontinuities over solutions with many small discontinuities. Some authors [9, 20] have proposed the use of completely general smoothness terms (for example the square of the disparity jumps between neighboring matches). Unfortunately, the use of completely general costs in full image formulations has lead to problems with NP-Complete complexity. In the work of [12] the authors first propose the use of fully general costs, but find that not only is the resulting problem NP-Complete, but even their subproblems are too. They therefore settle on a cost that only measures the existence of discontinuities, not their magnitude.

Our three camera algorithm uses a smoothness cost based on the absolute values of the disparity differences which are captured simultaneously with the occlusion penalty.

Algorithms and complexity Solving these minimization problems leads to algorithms with varying complexity. Formulations using algorithms based on dynamic programming and min-cut have polynomial efficiency, while the most general formulations give rise to NP-Complete problems. It remains an open problem to fully understand where the boundary lies between efficiently solvable formulations and intractable ones.

In this paper, we extend the bounds of solvable formulations by presenting polynomial time algorithms for minimal cost surfaces and show how they can be applied naturally to a three camera stereo problem.

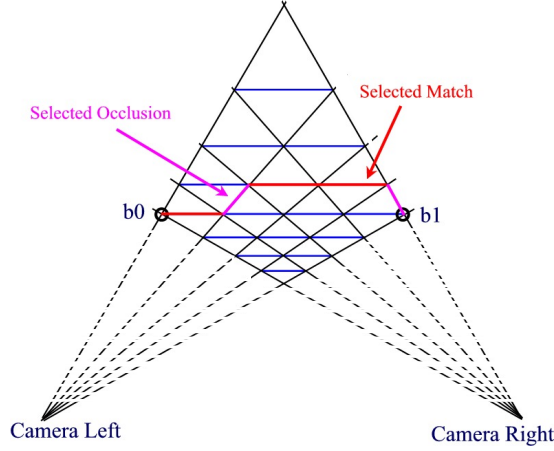


Fig. 1. Primal graph: Black edges are possible occlusion/disparity-discontinuity edges (pink when selected). Blue edges are possible match edges (red when selected).

Minimal Cost Surfaces Polynomial time algorithms for computing discrete spatial minimal cost surfaces appear to be absent from the computer science literature. In the crystallography literature, a construction almost identical to ours is described in [18].

One can interpret our minimal cost surface algorithm as an application of the voxel labeling algorithm of [17]. In their algorithm, one chooses one of two labels for each voxel in a spatial complex. A cost is used to describe the likelihood of a label being used for a specific voxel, and a smoothness cost is used to penalize neighboring voxels with differing labels. If one thinks of the two labels as *in front of* or *in back of* a separating surface, one can interpret the smoothness cost as a cost associated with the face separating two voxels.

Our work has also been very influenced by the construction of [15], which informally suggests the relationship between min-cut and minimal cost surfaces.

3 Polynomial time minimal cost surfaces

In this section we describe the discrete spatial minimal cost surface problem (MS), which we will later use to solve our stereo formulation. Since MS is a generalization of the planar shortest path (PSP) problem, we begin by describing it first. Then we will show how to solve the PSP problem as a min-cut problem. Of course, since more efficient shortest path algorithms exist, this would not be considered a practical approach for shortest path. But the more efficient algorithms do not generalize to higher dimensions while the min-cut approach does.

3.1 Planar shortest path

Problem formulation In the PSP problem, we are given as input an embedded planar graph consisting of vertices V , edges E and faces F (a CW-complex). We require that

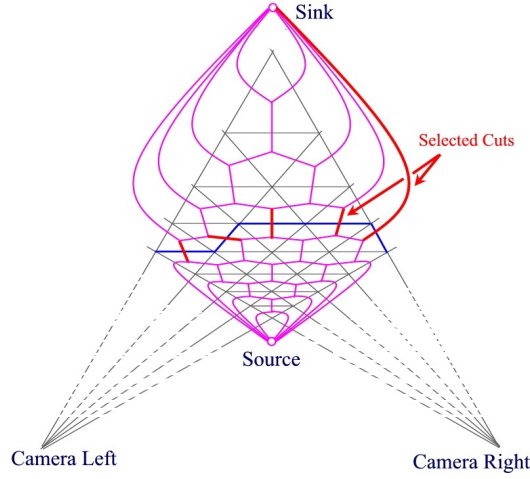


Fig. 2. Dual graph: Pink edges correspond to dual edges of the primal graph. Red edges are selected edges that cross the min-cut of this graph.

the complex has the disk-like topology, B^2 . We associate with each of the edges e_i a positive cost $c(e_i)$. Two special vertices on the exterior of the graph are labeled b_0 and b_1 . We denote by S , the set of curves made up of edges from E and whose boundary (zero-manifold) is made up of the vertices $\{b_0, b_1\}$. For each curve $s \in S$ we define its cost as the sum of its edge costs, $C(s) = \sum_{e \in s} c(e)$. The shortest path problem is to find the curve $s^* \in S$ with minimal cost. There are many efficient algorithms for the PSP problem, as well as for shortest paths in non-planar graphs [7].

Figure 1 shows a particular PSP problem that will be used below to solve a single scanline stereo problem.

Oriented PSP For our stereo formulation we will need to solve an *oriented* MS problem, so we describe here the oriented PSP problem. In an oriented PSP problem, we give each edge a *direction* (for example from right to left in Figure 1) and restrict solutions to directed paths. An equivalent representation that generalizes better to higher dimensions is to associate an orientation (normal vector) with each edge (e.g., with normals pointing towards the line connecting the camera centers in Figure 1), and require the solution path to have a consistent orientation.

Reduction to Min-cut The PSP problem can be solved in polynomial time by reduction to min-cut. First we construct the dual graph of the planar graph (Figure 2). That is, we associate with each face f_i a dual vertex \hat{v}_i . We also create a single *source* vertex \hat{r} and a single *sink* vertex \hat{k} . With each non-exterior edge e_i bounding two faces f_j and f_k , we associate a dual edge \hat{e}_i that connects \hat{v}_j and \hat{v}_k . The two boundary vertices b_1 and b_2 partition the exterior edges into two sets which we call upper U and lower L . For each exterior edge e_i in U , we associate a dual edge \hat{e}_i which connects the sink \hat{k} with the single face that e_i bounds. For each exterior edge e_i in L , we associate a dual edge \hat{e}_i which connects the source \hat{r} with the single face that e_i bounds. We set the *capacity* of each dual edge using the cost of the associated primal edge $c(\hat{e}_i) = c(e_i)$.

Def: A cut \hat{s} is a partition of the vertices \hat{v} into two sets \hat{R} and \hat{K} with $\hat{r} \in \hat{R}$ and $\hat{k} \in \hat{K}$

Def: The cost of a cut is the sum of the capacities of the edges between \hat{R} and \hat{K} ⁴.

Theorem: The edges across the min-cut \hat{s}^* are dual to the edges of the minimal cost path s^* .

A formal proof is beyond the scope of this paper but will appear in a forthcoming technical report. Informally the reason why this theorem is true is that a cut of the dual graph is a partition of the dual vertices into two sets. This cut then corresponds to a partition of the primal faces into two spatial regions ⁵. The dual edges across the cut correspond to the primal edges forming the boundary between these two spatial regions. Thus there is a natural duality between cuts and paths.

Directed cuts for oriented curves Given an oriented spatial graph, an associated directed dual graph can be built. The min-cut of this directed dual graph, will correspond to an oriented path in the oriented primal graph. The directed dual graph is created as follows. Given an oriented primal edge e_i with a normal specified by the input, we associate with it in the dual graph a pair of opposite facing directed edges. One edge, \hat{e}_i^+ in the direction of the specified normal, is assigned the capacity of the primal edge as before, $c(\hat{e}_i^+) = c(e_i)$. A second edge, \hat{e}_i^- in the direction opposite of the normal, is given an infinite capacity, $c(\hat{e}_i^-) = \infty$; this effectively eliminates it from possible min-cut solutions ⁶. Once again, the edges across the min-cut of this dual graph are dual to the minimal oriented path in the primal graph.

3.2 Minimal Cost Surface

The generalization of the PSP problem to higher dimensions is straightforward ⁷. In the discrete spatial minimal cost surface (MS) problem, we are given as input a CW-complex embedded in R^3 consisting of vertices, edges, faces and cells. We require that the embedded complex has the ball-like topology, B^3 . Associated with each face f_i is a cost $c(f_i)$. Also input is a closed curve B made up of a set of edges on the exterior of the complex. We denote by S , the set of surfaces made up of faces from F and whose boundary (one-manifold) is the curve B . For each surface $s \in S$ we define its cost as $C(s) = \sum_{f \in s} c(f)$. The problem is to find the surface s^* with minimal cost. In this paper, we describe a polynomial time algorithm for solving this problem.

In the *oriented* surface problem, each face has input with it an associated normal. The set of satisfactory surfaces must have a consistent set of normals.

⁴ When we deal with directed dual graphs for the oriented surface problem the cost will be the sum of the edges from \hat{R} to \hat{K} .

⁵ In a formal proof, one needs to show that each of these two spatial regions is simply connected and is consistent with the boundaries b_1 and b_2 . If there are zero cost edges in the primal graph, then there potentially can be a disconnected zero cost path loop in the minimal cost path. This is easy to detect and remove

⁶ If one simply removed these edges instead, then one could end up with a min cut solution whose associated primal edges do not form a valid path in the primal graph.

⁷ Unlike the minimal curve problem, which can be solved efficiently even for non-planar graphs, the minimal cost surface problem can only be solved efficiently for a complex that is embedded in three dimensional space. Without this restriction, the minimal cost surface problem becomes NP-Complete by reduction to min-cost planar triangulation [14].

The minimal cost surface can be found by reduction to min-cut. First we construct the dual graph of the spatial complex. That is, we associate with each cell c_i a dual vertex \hat{v}_i . We also create a single source vertex \hat{r} and a single sink vertex \hat{k} . With each non-exterior face f_i bounding two cells c_j and c_k , we associate a dual edge \hat{e}_i that connects \hat{v}_j and \hat{v}_k . The boundary edges B partition the exterior faces into parts which we call upper U and lower L . For each exterior face f_i in U , we associate a dual edge \hat{e}_i which connects the sink \hat{k} with the single cell that f_i bounds. For each exterior face f_i in L , we associate a dual edge \hat{e}_i which connects the source \hat{r} with the single cell that f_i bounds. We set the capacity of each dual edge equal to the cost of the associated primal face $c(\hat{e}_i) = c(f_i)$.

Theorem: The edges across the min-cut \hat{s}^* are dual to the faces of the minimal cost surface s^* .

4 Two-Camera Stereo By Minimal Cost Surface

The minimal cost surface formulations can be used to solve a number of minimization problems that arise in stereo vision. We will describe a sequence of three problems. We start with a scanline formulation for two camera stereo just to establish the basic idea. Then we will describe a full image formulation for two camera stereo. This MS problem will reduce to a min-cut problem quite similar to that of Ishikawa and Geiger [11]. Finally we describe a novel three camera stereo formulation.

4.1 Two camera single scanline

For a “single scanline two camera” problem, we can set up the following formulation. A stereo solution is described by a set of matching pairs of pixels from the left and right cameras. We enforce uniqueness; a pixel can be matched at most once. We also enforce a monotonicity/ordering constraint on these matches. A match between a left and right pixel costs us some amount based on the inverse of the likelihood that the two pixels are images of the same point in space. For example, we can use the squared difference of their colors or a more complex metric that examines correlations of larger windows. An unmatched pixel costs us some fixed amount⁸. We define two matches to be neighbors if either their left or right pixels are adjacent. We impose a smoothness penalty on two neighboring matches with differing disparities that are both in the solution which is simply the absolute value of the difference between the two posited disparities.

We can solve for the global minimum of this problem by solving for the shortest path in the planar graph shown in Figure 1. Each quadrilateral grid cell represents the intersection of the projective extrusions of two pixels. (We think of each pixel as an interval in its scanline, not as a point). We bisect each quadrilateral with a cross edge (shown in blue) projecting to exactly a one pixel-interval in each of the two images. These “match edges” represent possible pixel matches. The cost of these edges is based on the observed pixel differences. The black edges, created by projecting the pixels’ boundaries, project into pixel-intervals in one image and no pixels (an interpixel point) in the other image. These edges represent changes in disparity between matches and

⁸ The cost of an unmatched pixel may be derived using Bayesian methods [9, 3], and can even be altered based on the “edginess” of that image region.

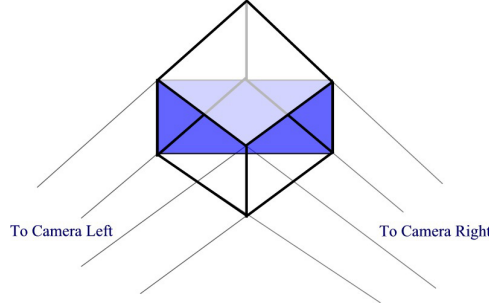


Fig. 3. The intersection of the extrusions of one rectified pixel from each of two cameras forms a cuboidal voxel. The blue quadrilateral represents a possible match surface. All outer faces of the voxel represent possible disparity discontinuities.

thus carry the cost of the smoothness penalty. They also represent an unmatched or occluded pixel. With all edges, we associate normals that face the line connecting the two camera centers. If we impose both the uniqueness and monotonicity constraint, then a valid stereo solution corresponds to an oriented path in this graph between the outer boundary b_0 and b_1 . The cost of the path will be the sum of the match costs plus the smoothness cost (the count of the number of occluded (pink) edges chosen).

This shortest path problem can be solved using Dijkstra’s algorithm, but for explanatory reasons, we will reduce it to a min-cost problem. We do this by building the dual graph, adding a source and sink vertex as shown in Figure 2, and then solving for the min-cut. One should note the similarity of this graph with Figure 3 from [11]. The main reason for the differences between this graph and that of [11] arises from the inclusion of tilted plane (non-unique matching) solutions allowed in their formulation.

4.2 Two camera full image

We can use the minimal cost surface construction described above to solve for a two camera full image problem. We build a spatial complex for the minimal cost surface problem which is effectively a 3D extrusion of the planar graph of Figure 1. In this construction, the costs used in this problem are the same as the single scanline formulation above, except that vertical neighbors are also considered for the smoothness penalty.

More specifically, we associate pixel values with quadrilateral pixel regions of their image planes (not single points). These pixel regions are projectively extruded to create a 3D voxel grid. Each of these voxels is bisected with a *match* quadrilateral drawn in blue in Figure 3. The blue quadrilaterals represent matches and are priced according to the color differences of the two associated pixels. The four vertical faces are priced to represent occlusion and smoothness costs. The two horizontal faces are priced to represent the smoothness costs⁹. We define as the boundary B , the leftmost edge of the complex, the rightmost edge of the complex, and any arbitrary connecting paths on the upper and lower exteriors of the complex¹⁰.

⁹ The horizontal faces on the upper and lower exterior are given zero cost, since there are no neighboring pixels to measure smoothness.

¹⁰ These can be arbitrary since the upper and lower exterior faces have zero cost.

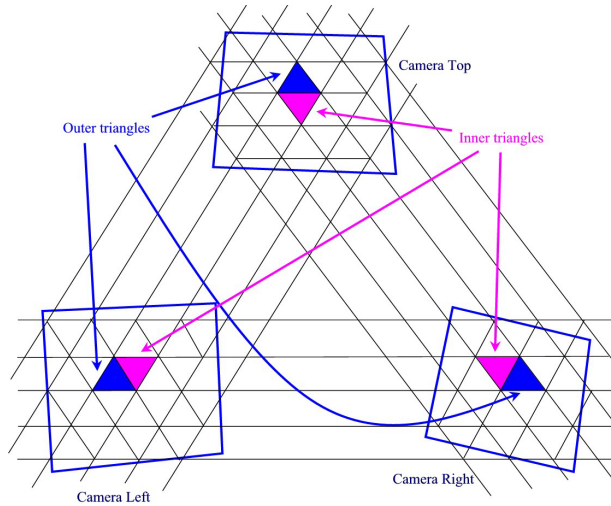


Fig. 4. Pairwise rectification of three images. Rectified quadrilateral pixels are divided into two triangular (inner and outer) subpixels.

Once again, we solve this problem by solving for a min-cut of the appropriate dual graph. This graph again has strong similarities to the graph in Figure 5a of [11].

5 Three-camera stereo by minimal cost surface

In this section we show how a minimal cost surface problem can be used to solve a three camera, full image formulation. We do not assume any special relationship between the cameras, nor choose any one to serve as a base, but rather treat them symmetrically.

To help keep the bookkeeping tractable, we first perform trinocular rectification [1] of the images (see Figure 4). To accomplish this, we find the trifocal plane passing through the three camera centers and reproject the pictures onto an image plane parallel to this trifocal plane. Between each pair of images, there is a set of epipolar lines parallel to the line connecting those two cameras' centers. This defines two sets of lines in each image, resulting in a quadrilateral pixel grid on each image. Each quadrilateral pixel is then bisected to create two triangular subpixels, an inner (pink) and outer (blue) triangle as shown in Figure 4. We then resample the images with one color value per triangle. The reason for this bisection will be described below. This rectification does not introduce a large amount of distortion when the cameras are configured to be roughly facing in the direction perpendicular to the trifocal plane; but degenerate configurations are possible, such as when the cameras face towards the trifocal plane.

This rectification has the following nice property, if one takes one quadrilateral pixel from each of the three images, and projectively extrudes these pixels into three-space, and then takes the intersection of these three solids, this intersection will either consist of the empty set or it will consist of a cuboidal voxel (see Figure 5). This voxel has 6 quadrilateral faces. Each of these quadrilateral faces projects back as a quadrilateral pixel in one image, and as an edge between two quadrilaterals in the other two images. This voxel grid forms the basis for our minimal cost surface problem.

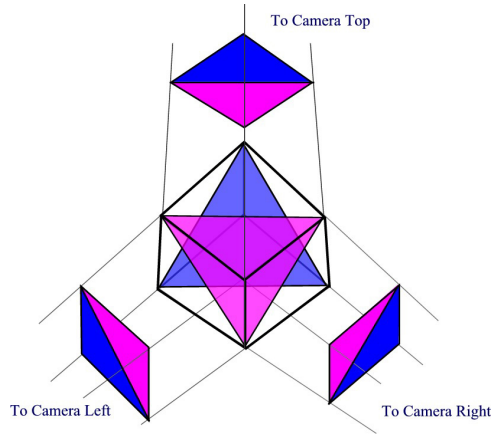


Fig. 5. The intersection of the projection of three rectified pixels forms a cuboidal voxel. The pink triangular face projects in the three images as inner triangular subpixels forming a possible *front* match surface. The blue triangular face projects in the three images as outer triangular subpixels forming a possible *back* match surface. All the external triangular faces of the cuboid are possible occlusion surfaces.

5.1 Three-way match surfaces only

In the simplest three camera formulation, we introduce match surfaces that are seen in subpixels of all three cameras. In other words, we will not consider matches between only two of the three cameras.

In the scanline and 2 camera full image formulations, we introduced bisecting edges and faces respectively into the graph which projected as pixels in both images. In this formulation we introduce two bisecting triangle faces into each voxel (see Figure 5), a front and rear triangle face. A front triangle face has the property that it projects to inner triangle subpixels in all three images. The rear triangle face projects to three outer triangular subpixels. These front and rear triangle faces trisect the voxel into three cells we call front, middle and back (see Figure 6). The edges of these two triangles faces bisect the 6 quadrilateral faces of the voxel into 12 triangular faces.

The front and rear triangle faces in each voxel represent possible simultaneous three-camera-matches. The match cost for each triangle face is based on the color differences of the three triangular subpixels that it projects to.

The outer faces of a voxel represent unmatched subpixels, and are assigned an occlusion cost. Just like in the single scanline formulation, this occlusion cost also accounts for a smoothness term, since disparity changes can only occur along with occlusions.

As in the two camera case, in the three camera formulation we enforce a monotonicity constraint that states that 3 features seen in all three images must have the same clockwise ordering. Enforcing the uniqueness and monotonicity constraints implies that a valid stereo solution can be represented as a single oriented surface in this complex. The orientation of all these 14 triangle faces in a voxel is given by the normal that points towards the trifocal plane.

This stereo formulation can be solved as a minimal cost surface problem over the complex we have described. To derive an efficient solution for the minimal cost surface,

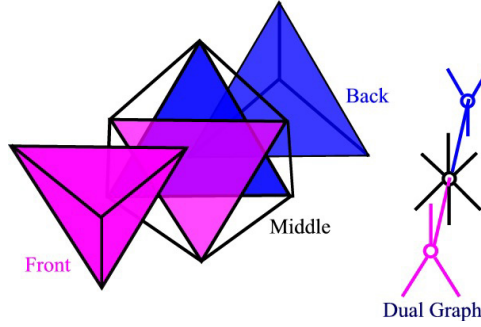


Fig. 6. The front and back match triangles subdivide the voxel into three cells. In the dual graph, each cell has an associated vertex.

we once again construct the dual graph to this spatial complex. We create one vertex dual to each volumetric region. In particular, there are three vertices dual to the three cells within each cuboidal voxel; one vertex dual to the tetrahedron in front of the front triangle, one dual to the tetrahedron behind the back triangle, and one dual to the middle region bounded by the two triangles (see Figure 6). These three dual vertices are connected by dual edges: front-middle and middle-back. Dual vertices from neighboring voxels are connected with dual edges piercing each of the 12 outer triangular faces of the voxel. The *capacity* of each edge is set to the costs associated with the faces they cross. The two edges within a voxel (pink and blue in Figure 6) represent the two possible matches. The 12 other (black) edges represent occlusions.

5.2 Adding two-way match surfaces

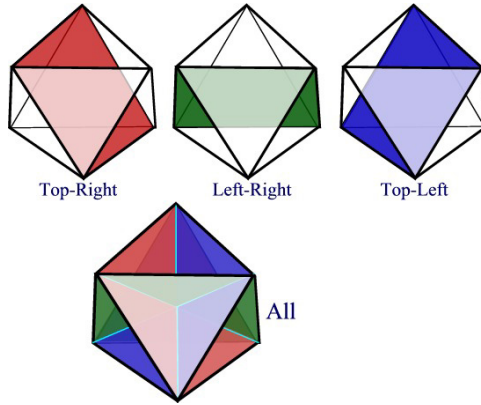


Fig. 7. The middle cell of Figure 6 is divided up into 8 subcells using 3 planes.

In a solution to the “three-way matches only” formulation, a (triangular) subpixel in one image either matches exactly one pixel in both of the other two images, or matches

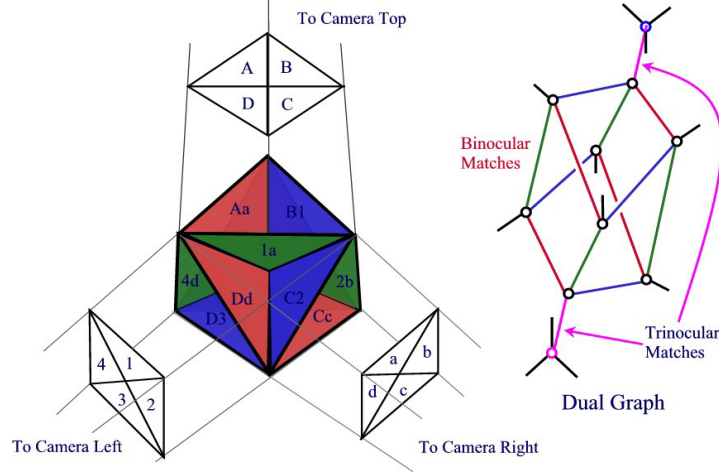


Fig. 8. Left: The middle cell of a cuboidal voxel in the two-way formulation. Right: The dual graph of a cuboidal voxel in this formulation.

no pixels in either of the two other images. By creating a more complicated spatial complex, we can allow the solution to also include surface elements that project with non zero area to only two out of the three images.

This is done as follows. The front and back cells of the cuboidal voxel in Figure 6 remain intact. We then introduce three new diagonal (2-camera) planes dividing the middle cell into 8 tetrahedra as shown in Figure 7. Each of the three introduced planes has been split up into 4 triangular faces by the other two planes. Each of these triangular faces project as one half of an inner or outer triangular subpixels in two cameras (see Figure 8). Thus, in our rectified images, we now consider each original quadrilateral pixel as being decomposed into four triangular subpixels. Weights for these faces are described in the discussion section below. The minimal cost surface is found by reduction to min-cut. The dual graph of one entire cuboid is visualized on the right side of Figure 8.

5.3 Bounds

In the three rectified images (see Figure 4) we index pixels as i, j (left to right, bottom to top) in camera left, j, k in camera right (bottom to top, right to left) and k, i in camera top (right to left, top to bottom). Cuboidal voxels are then indexed as i, j, k .

Not all possible voxels will fall completely in the three fields of view. Those outside are discarded. The user can also set minimal and maximal allowed (positive) disparities. The disparity range can be conservatively enforced by testing each voxel using appropriate affine expressions of $i + j + k$. Voxels not satisfying these constraints are also discarded. The remaining voxels make up our volume of interest.

A boundary curve B must be chosen to partition the external faces into two sets L and U . The dual external edges will connect to the source and the sink respectively. We generalize the boundary selection used in the single scanline formulation to select

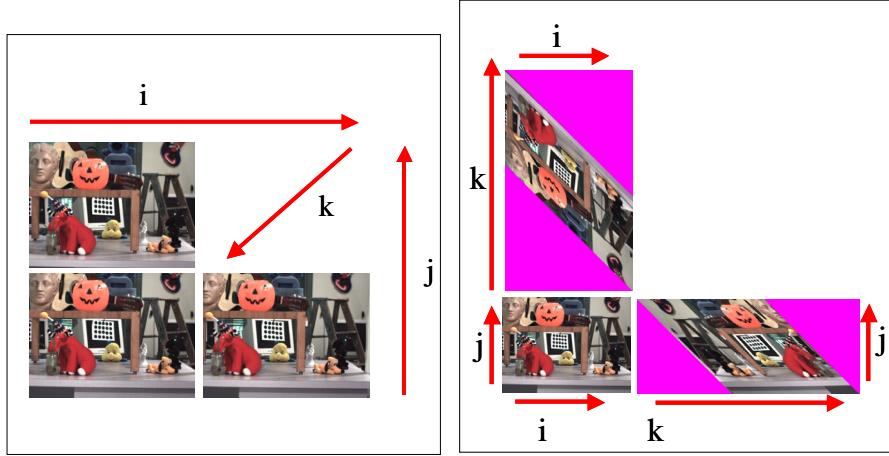


Fig. 9. Rectification. Left: three images as seen on trifocal plane. Right: resampled and indexed by i, j, k .

the boundary here (see Figure 1). We first find the plane parallel to the trifocal plane which has the maximal area intersection with the volume. All external faces that are completely in front of this plane are placed in L , while the others are placed in U . This maximal plane test allows us to include the most possible three-camera matches in the solution.

5.4 Complexity

If there are n pixels in each of the three images and we restrict our solutions to d possible disparity levels, then the spatial complex has nd voxels. With two-way faces included, this results in $10nd$ cells and $20nd$ faces; the dual graph has correspondingly many vertices and edges. Using known flow algorithms [5], this leads to a worst case time complexity of $O((nd)^2 \log(nd))$.

6 Implementation and Results

We have coded our minimal cost surface stereo algorithm and tested it on a number of datasets.

An example of our rectification is shown in Figure 9. On the left is the three original images. On the right we show the resampled trinocularly rectified images. Common features in any pair of images must agree on any shared coordinates (i , j , or k). In our implementation, we calculate only a single color for each quadrilateral pixel; we do not compute separate values for the triangular subpixels.

We used the publicly available PRF library to solve our min-cut problems [6]. To calculate color similarities for the match edge costs, we used the method of [4]; use of this method greatly increased the quality of the results. The occlusion weights were set by a user controlled parameter. The algorithm was run with two-way matches included,

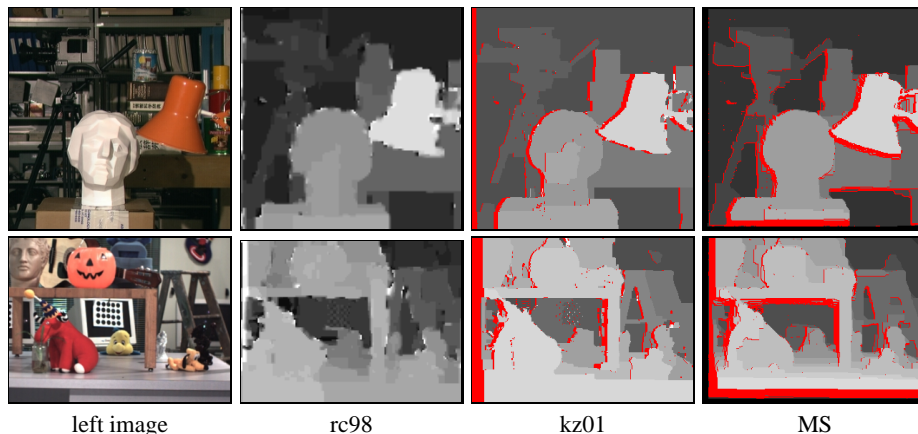


Fig. 10. Comparison of this algorithm to rc98 and kz01 on two datasets. Unmatched pixels drawn in red. Note: the gray levels between the different algorithms are not calibrated.

on a 1GHz Pentium III, running Windows. Execution times were on the order of 1-5 minutes.

Figure 10 shows our results and compares them to those obtained using two other algorithms. The datasets used were the Tsukuba dataset, and the toys data set from [10]. The Tsukuba dataset was cropped to 256 by 256 pixels to alleviate physical memory limitations. The toy dataset images are 310 by 210 pixels. In both datasets, we used the minimum baseline spacing giving us roughly 10-12 disparity levels. We used these data as input for the publicly available implementations of rc98 [15] and kz01 [12]. As input to our algorithm we used triplets of images in an “L” configuration with both vertical and horizontal parallax. As input to the other two codes, we used a left-right stereo pair. We adjusted their parameters attempting to produce the best results we could. In Figure 10, we show the (lower) left image of the input. The outputs are shown as grayscale disparity maps. Unmatched pixels are drawn in red. In our algorithm a pixel is made up of 4 subpixels; for simplicity, we colored the pixel by the disparity of the frontmost three-way matching subpixel. Pixels that had only occlusion or two-way faces were drawn in red.

The MS algorithm appeared to give better results than rc98 for both datasets. MS seemed to do about the same as kz01 on the Tsukuba dataset and slightly better than kz01 on the toys data set (especially on the receding plane of the table top).

7 Discussion

In this paper we have described an efficient method for solving for minimal cost surfaces in a 3D spatial structure. We have shown how this can be applied to a novel three camera stereo formulation.

Optimally, one could use the two-way faces to model surfaces that are seen in two views but occluded in a third view. Unfortunately, an oriented surface in our complex cannot contain any regions with depth complexity greater than one as seen from any of

the three views. Thus an oriented surface cannot represent such partially hidden layers. Therefore, we use the two-way faces not to represent matches, but as another way of representing occlusions; we price them as occlusions, and display these faces as red in the output. In practice we found that the inclusion of these faces greatly improved the results simply because they allow the minimal cost surfaces to have more directional flexibility, and provided a more accurate measurement of the occluded “surface area”.

In future work we would like to explore algorithms that enforce uniqueness, but allow for true two camera matches at depths that are occluded in a third view. Future work also includes gaining a better understanding of where the boundary between NP-Complete and polynomially solvable stereo problems lies.

References

1. N. Ayache and C. Hansen. Rectification of images for binocular and trinocular stereovision. *Proc. International Conference on Pattern Recognition 1998*, pages 11–16.
2. H. Baker. *Depth from edge and intensity based stereo*. PhD thesis, University of Illinois at Urbana Champaign, 1981.
3. P. Belumeur and D. Mumford. A bayesian treatment of the stereo correspondance problem using half-occluded regions. *IEEE CVPR '92*.
4. S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE PAMI*, 20(4):401–406, 1998.
5. C. Chekuri, A. Goldberg, D. Karger, M. Levine, and C. Stein. Experimental study of minimum cut algorithms. *Proc ACM SODA*, 1997.
6. B. Cherkassky and A. Goldberg. Prf library from www.star-lba.com/goldberg/soft.html.
7. T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, 2001.
8. I. Cox, S. Hingorani, B. Maggas, and S. Rao. A maximum likelihood stereo algorithm. *Computer vision and image understanding*, 63(3), 1996.
9. D. Geiger, B. Landendorf, and A. Yuille. Occlusion and binocular stereo. *IJCV*, 14:211–226, 1995.
10. Aaron Isaksen. toys dataset by request.
11. H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. *Proc. ECCV 98*.
12. V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. *Proc. ICCV 2001*.
13. Y. Ohta and T. Kanade. Stereo by intra and inter scanline search using dynamic programming. *IEEE PAMI*, 7(2):139–154, 1985.
14. J. O’Rourke. *Computational Geometry in C*. Cambridge University Press, 1994.
15. S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. *Proc. ICCV 1998*.
16. S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. *IJCV*, 35(2):151–173, 1999.
17. D. Snow, P. Viola, and R. Zabih. Exact voxel occupancy with graph cuts. *Proc. IEEE CVPR 2000*.
18. J. Sullivan. *A Crystalline Approximation Theorem for Hypersurfaces*. PhD thesis, Dept. of Mathematics, Princeton University, 1990.
19. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Proc. ICCV 1999*.
20. Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. *Proc. IEEE CVPR 1998*.